(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 May 2001 (10.05.2001)

PCT

(10) International Publication Number
WO 01/33778 A1

(51) International Patent Classification[7]: H04L 12/56, H04Q 11/04

(21) International Application Number: PCT/IL00/00581

(22) International Filing Date: 19 September 2000 (19.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
132694 1 November 1999 (01.11.1999) IL

(71) Applicant *(for all designated States except US)*: TER-ACROSS LTD. [IL/IL]; P.O. Box 3030, Industrial Park, 84965 Omer (IL).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: BARZILAI, Ehud [IL/IL]; P.O. Box 3030, Industrial Park, 84965 Omer (IL).

(74) Agent: REINHOLD COHN AND PARTNERS; P.O. Box 4060, 61040 Tel-Aviv (IL).

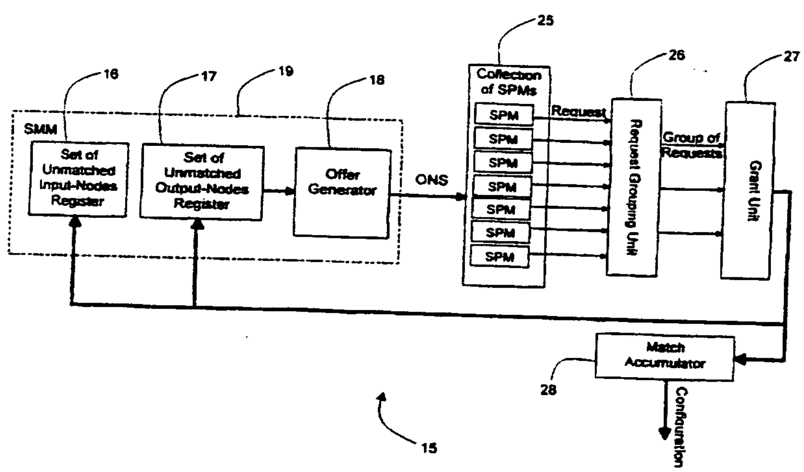
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR HIGH-SPEED, HIGH-CAPACITY PACKET-SCHEDULING SUPPORTING QUALITY OF SERVICE IN COMMUNICATIONS NETWORKS



(57) Abstract: A method and scheduler (15) for scheduling data packets transported from input-nodes (11) to output-nodes (12) via a cross-connect fabric (13) having C channels, the data packets being associated with a set of N input-nodes each having a plurality of M queues each for queuing data packets for routing to a corresponding one of M output-nodes. A priority value is associated with each queue in each input-node, and a snapshot is taken of queue priorities. Sets of available input-nodes and available output-nodes are received which may initially contain all input-nodes and output-nodes, respectively, and a subset (ONS) of the set of available output-nodes is selected. For each input-node one request is submitted containing an identity of a requested output-node in the ONS and a corresponding priority value. Requests are grouped according to the identity of the requested output-node, and the output-node associated with each group is matched with the input-node having the highest priority request in the respective group. The matches are accumulated and matched input- and output-nodes are removed from the respective sets of available input- and output-nodes, the whole process being repeating as required.

WO 01/33778 A1

Method and apparatus for high-speed, high-capacity packet-scheduling supporting quality of service in communications networks

FIELD OF THE INVENTION

The present invention relates to the field of communication networks, and particularly to real-time packet scheduling in packet switched networks.

BACKGROUND OF THE INVENTION

As Internet traffic volume increases at an exponential rate, the search for high-performance and scalable packet-switching technologies is broadening. Traffic passing through the Internet is not only increasing in volume but also becoming more demanding in terms of quality of service (QoS). Examples of QoS parameters are packet delay, packet delay variation and packet loss. Existing and emerging multimedia applications, such as voice and video, which are growing more prevalent, require strict channel characteristics in order to function properly.

Broadband network infrastructures are coarsely composed of two basic building blocks: (1) high-speed point-to-point links and (2) high-performance network switching devices. While reliable high-speed point-to-point communications have been demonstrated using optical technologies, such as Wave Division Multiplexing (WDM), switches and routers that can efficiently manage extensive amounts of diversely characterized traffic loads are not yet available. Hence, reduction of the bottleneck of communication network infrastructures has shifted towards designing such high-performance switches and routers.

It is generally acknowledged that the two main goals of network switches are 1) to utilize the available internal bandwidth optimally while at the same time 2) support QoS requirements. Constraints derived from these goals typically contradict in the sense that maximal bandwidth utilization does not necessarily mutually correlate to the support of the most urgent traffic flows. This concept has spawned a vast range of scheduling adaptation schemes, each seeking to offer high capacity, large number of ports and low latency requirements.

Many of these schemes employ output-queuing mechanisms, which means that packets (ATM, IP or any other type of packets) arriving at the input-node are transmitted through the cross-connect fabric to designated queues at output-nodes. In order to overcome collision in an *N-by-N* cross-connect fabric, either N^2 independent channels or circuitry capable of switching packets N times faster than the fastest input port's speed must be employed. Considering today's high line rates, N times faster circuitry is infeasible. Internal links are valuable resources making the realization of N^2 such links wasteful and infeasible.

Typical designs apply either centralized-queuing or output-queuing mechanisms in order to maximize switch bandwidth. However, as line rates and port densities increase, output queuing is found impractical.

An alternative to output-queuing is input-queuing, whereby cell buffering is managed at the switch input stage. It is well known that an input-queued switch employing a single FIFO at each input-node may achieve a maximum of 58.6% throughput due to the head-of-line (HOL) blocking phenomenon. A well-practiced technique, which entirely eliminates the HOL blocking, is *Virtual Output Queuing* (VOQ). In VOQ each input-node maintains a separate queue for each output. Arriving packets are classified at a primal stage to queues corresponding to the packet's designated destination. Such information is typically available within the packet header. In general, the goal of a scheduling mechanism is to determine, at any given time, which queue is to be served, i.e. permitted to transfer packets to its destined output.

Several scheduling algorithms have been proposed for VOQ switches. Most high-performance algorithms known to date are too complex to be implemented in hardware and are found unsuitable for switches with high port densities and high line rates. Moreover, the algorithms proposed are commonly evaluated under uniform traffic conditions, which clearly does not represent real life traffic. As the traffic becomes less uniform, these algorithms usually suffer from severe performance degradation. One method of enhancing VOQ based switching is to increase the internal "speedup" of the switch. A switch with a speedup of L can transport L packets to any single output-node in one cell-time (the time period in which one cell arrives at the fastest input port). However, the switching-core speed is a paramount resource limited by available technology, making speedup a drawback of any scheduling approach. In order to support QoS, VOQ is frequently expanded by assigning r different queues for each destination (as opposed to just one), whereby each queue corresponds to a distinct QoS class. Contention for transmission is thus carried out not only among queues in different input ports relating to the

same destination port, but also among different class queues in any single input port designated for the same destination.

Although known scheduling algorithms focus on packets of fixed length, many network protocols, such as IP, have variable length packets. Most switching engines today segment these packets into fixed-length packets (or "cells") prior to entering the switch fabric. The original packets are reconstructed at the output stage. This methodology is commonly practiced in order to achieve high performance. Accordingly, the methods described here may apply to both fixed and variable length packets.

Currently deployed scheduling algorithms practice some variation of a Round Robin scheme in which each queue is scanned in a cyclic manner. These schemes suffer from many disadvantages, including deficient support of global QoS provisioning and limited scalability with respect to line speeds and port densities. The latter is an extreme weakness of these schemes owing to the demand for connectivity of order N^2, where N denotes the number of ports in the switch. As a result, switch resources are not optimally exploited yielding limited switching performance.

Other methods carry out more sophisticated scheduling approaches, which better exploit the switch resources. Still, these methods are complex and require relatively long processing periods, thus limiting the supported data rate, since decisions related to optimal scheduling are not produced in real-time.

It would therefore clearly be desirable to provide a fast, real-time, scalable, high-capacity packet scheduling solution which supports QoS in high-speed packet switched networks.

It is therefore an object of the present invention to provide a method and apparatus for the scheduling of data packets in packet-switched networks, wherein the above-mentioned drawbacks are reduced or eliminated.

It is another object of the invention to provide a method and apparatus for the scheduling of data in packet-switched networks, which is carried out in real-time.

It is yet another object of the invention to provide a method and apparatus for the scheduling of data in packet switched networks, which more effectively and adaptively exploits the system resources while supporting dynamic QoS provisioning.

Other objects and advantages of the invention will become apparent as the description proceeds.

SUMMARY OF THE INVENTION

This object is realized in accordance with a first aspect of the invention by a method for scheduling data packets transported from input-nodes to output-nodes via a cross-connect fabric having C channels, said data packets being associated with a set of N input-nodes each having a plurality of M queues each for queuing data packets for routing to a corresponding one of M output-nodes, said method comprising the steps of:

(a) associating a priority value with each queue in each input-node, and making a snapshot of queue priorities,

(b) receiving sets of available input-nodes and available output–nodes which may initially contain all input-nodes and output–nodes, respectively,

(c) selecting a subset (ONS) of the set of available output-nodes,

(d) submitting one request for each input-node, said request containing an identity of a requested output-node in the ONS and a corresponding priority value,

(e) grouping requests according to the identity of the requested output-node,

(f) matching the output-node associated with each group with the input-node having the highest priority request in the respective group,

(g) accumulating matches and removing matched input- and output–nodes from the respective sets of available input- and output–nodes, and

(h) repeating steps **(c)** to **(g)** as required.

According to a second aspect of the invention there is provided a scheduler for scheduling data packets transported from input-nodes to output-nodes via a cross-connect fabric having ***C*** channels, said data packets being associated with a set of ***N*** input-nodes each having a plurality of ***M*** queues each for queuing data packets for routing to a corresponding one of ***M*** output-nodes, said scheduler comprising:

an available input-nodes register and an available output-nodes register for storing data representative of respective sets of available input–nodes and available output–nodes,

an offer generator coupled to the available output-nodes register for selecting a subset (ONS) of the set of available output-nodes,

a collection of source-port modules (SPMs) each containing at least one register for storing a snapshot of queue priorities, said SPMs being coupled to the offer generator for submitting requests containing an identity of a requested output-node in the ONS and a corresponding priority value,

a request grouping unit coupled to the SPMs for grouping requests according to the identity of the requested output-node.

a grant unit coupled to the request grouping unit for matching the output-node associated with each group with the input-node having the highest priority request in the respective group, and

a match accumulator coupled to the grant unit for accumulating matches and removing matched input- and output–nodes from the respective sets of available input- and output–nodes.

The present invention is directed to a method for real-time scheduling of data packets in high-speed, high capacity packet switches. The invention is further directed at a scheduling scheme that utilizes maximal bandwidth while supporting QoS requirements. It is assumed that the packet switch consists of input-nodes, output-nodes and a switching fabric capable of connecting an input-node to an output-node for the purpose of transporting packets.

Most switches today have bi-directional ports capable of simultaneously transporting data to/from the switch. In such a case every switch port contains one input-node and one output-node, and the number of input-nodes is equal to the number of output-nodes. Still, the invention is not limited to such a case and can be applied to a switch with any number of input-nodes and any number of output-nodes. For the purpose of this description, however, it will be assumed that there are N input-nodes and N output-nodes in the switch.

It is further assumed that there are C internal transmission channels in the switch cross-connect fabric. Each channel can be used to connect one input-node to one output-node at any given time. Although in typical realizations C equals to N, this is not obligatory. In addition, the cross–connect may be based on optical, electrical or any other physical media.

A collision occurs when two or more input-nodes need to transmit to the same output-node. It is the object of the scheduler, and hence of the invention, to provide a method and apparatus for determining which input-node will be assigned a channel to which output-node, in accordance with optimal utilization of the switch resources (channels) and maximal compliance with QoS requirements.

Data packets arrive via the switch port and are queued in multiple queues at the input-node. At any given time, a priority value is determined for each queue in each input-node based on packet arrival statistical information and the queue's identity (e.g. the output-queue for which cells

in the queue are destined, and/or a fixed QoS-class associated with the queuc). Examples of statistical information used for determining priority may be the number of packets occupying the queue and their respective arrival times.

In the invention, the matching of input-nodes to output-nodes via channels is attained by conducting a sequence of output-node contentions whereby all previously unmatched input-nodes contend for a given set of available (i.e. unmatched) output-nodes. At the end of each said contention, input-output-node matches are established based on maximum priority matching. A complete matching configuration is achieved at the end of this sequence of contentions.

By contending for a subset of the available output-nodes in each stage, speed, fairness and efficiency are gained. As opposed to common scheduling schemes, the invention does not require N^2 internal channels nor does it require N-times speedup to achieve near 100% utilization.

The invention may be implemented utilizing a pipelined architecture to accelerate the completion of the matching process. In this manner, one contention cycle's execution is broken into several stages executing concurrently, with the output of each stage being fed to the input of its successor.

The matching procedure is based on priority values produced by a queue-tracking mechanism, tracking each of the queues in the input-nodes. Such priority values can be calculated prior to the series of contentions according to different parameters, such as internal packet information, length of formed queues, packet waiting times, packet arrival rates and desired QoS. A plurality of comparator units, organized in a hierarchical structure, allow for fast finding of the maximal priority value.

The invention is also directed to an apparatus for real-time packet scheduling in high-rate, high port density packet-switched networks supporting QoS, comprising circuitry for globally determining packet

scheduling according to information on priorities and available transmission resources at all switch nodes. The apparatus includes a switch with a plurality of input- and output-nodes each having respective incoming and outgoing data links; a switching fabric capable of transporting data from input- to output-nodes via assigned channels; and switching control circuitry (Scheduler) for controlling this switching fabric. The scheduler controls the channel assignment procedure at each node via designated control lines.

The invention can also be applied to inter-processor communications in the field of computer architectures. In such applications the switch is analogous to a multi-processor computer, the switch ports are analogous to the processors in this computer, and packets are analogous to batches of information exchanged among the processors. The switching fabric is analogous to whatever mechanism is used to transport information from one processor to another (possibly a network of information buses).

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other characteristics and advantages of the invention will be better understood through the following illustrative and non-limiting detailed description of the preferred embodiments thereof, with reference to the appended drawings, wherein:

Fig. 1 is a block diagram of a switch, used for high-speed, high-capacity packet-switched data communications, according to a preferred embodiment of the invention;

Fig. 2 is a block diagram showing functionally the switch scheduler architecture shown in Fig. 1;

Fig. 3 is a block diagram showing a detail of the source-port module shown functionally in Fig. 2;

Fig. 4 is a block diagram of a hardware implementation of a comparison tree for determining the maximal priority value among queues

corresponding to members of the ONS in the source–port module shown in Fig. 3;

Fig. 5 is a flow diagram showing the principal steps performed by the scheduler shown in Fig. 2; and

Fig. 6 is a flow diagram showing the principal steps performed by the SPM for generating a request.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Fig.1 is a block diagram of a switch shown generally as 10, which comprises N input-nodes 11 and M output-nodes 12, used for high-speed, high-capacity data transfer according to a preferred embodiment of the invention. As mentioned above, in most switches M=N as input-nodes and output-nodes are paired to form bi-directional ports, but the invention is equally applicable to the case where $M \neq N$.

Packets are transferred from input- to output-nodes via a cross–connect fabric 13. The cross–connect fabric 13 may be optical, electrical or based on other physical media. A switch scheduler 15 is responsive to data relating to incoming traffic and queuing statistics for establishing input-output-node matching decisions to be used by the nodes and cross-connect fabric for transferring packets within the switch.

Fig. 2 shows functionally the scheduler 15. Registers 16 and 17 store a set of unmatched input-nodes and a set of unmatched output-nodes, respectively. The register 17 is coupled to an offer generator 18 that selects a subset of available (i.e. unmatched) output-nodes in respect of which unmatched input-nodes are to be selected. The registers 16 and 17 together with the offer generator 18 constitute a scheduling management module 19, which thus generates a subset of the unmatched output-nodes referred to as the "Offered Nodes Set" (ONS) over which to contend. The ONS may be derived by randomly selecting a size-limited subset of the available (i.e.

unmatched) output-nodes. Alternatively, the subsets may be selected in a sequential manner out of the set of the available output-nodes.

The ONS is fed to a collection of source-port modules (SPMs) 25, each associated with a respective input-node in the switch and described in further detail below with reference to Fig. 3 of the drawings. The output from each SPM 25 is fed to a Request Grouping Unit 26, which partitions the collection of requests into groups. Each such group contains all requests made by input-nodes for one specific output-node. Since requests can only be made for members of the ONS, the number of such groups cannot exceed the number of members in the ONS.

Although not limited to such a queuing scheme, the packets are queued in a VOQ realization. In order to support QoS provisioning, each queue may be associated with both an output-node and a QoS class. The SPM 25 maintains logging of coherent statistical data regarding the arrival of packets to each of the queues in the node. Such information includes, but is not limited to, the number of packets occupying each queue and their arrival times. It is another task of each SPM to associate with each queue a priority level, which is based on the logged statistical data, and is recalculated continuously or when needed. The priority generating mechanism should be kept identical in all SPMs if global fairness is to be assured, although the manner in which priorities are determined is not itself a feature of the invention. From here on, the priority level of queue i in node j will be denoted by P_{ij}.

Each one of the groups of requests is then passed to a Grant Unit 27. The Grant Unit 27 determines (concurrently for all groups) one prevailing input-node for each group of requests, by examining the requested priorities and selecting the request having the highest priority in the group. Since each group corresponds to one known output-node (from the ONS), and the prevailing request was made by one known input-node, a match may then be formed between these input- and output-nodes. To this end, a Match

Accumulator 28 is responsively coupled to the Grant Unit 27 via a bus 29 bearing the respective identities of the matching input- and output-nodes.

Fig. 3 shows functionally a detail of the SPM 25 associated with each input-node in the switch. Incoming data packets are stored at each input-node in an array of queues. Within the SPM 25 corresponding to each input-node, a Queue Tracking Mechanism 35 tracks packet arrival to and departure from the corresponding input-node and subsequently generates an array of priority values each in respect of a single queue within the input-node. The priority values are fed to an output-node filter logic unit 36 whose output is a set of requested output-nodes each having a respective priority or a status signal indicative of there being no requested output-nodes, indicated by the priority being equal to zero. In this case, all output-nodes in the offered set are matched with the corresponding input-node having the highest priority request, and the data at each input-node is now routed to its destination. To this end, a maximal priority circuit 37 coupled to the output-node filter logic unit 36 outputs a single input-node having the highest priority request.

Fig. 4 shows an exemplary configuration of the maximal priority circuit 37 in a 4-input-node switch with two QoS classes per output-node. A plurality of comparator units 40 are disposed at the SPMs 25 and at the Grant Unit 27 and are organized in a hierarchical structure so as to determine the maximal priority value. In each SPM, the maximal priority value is to be found from among the priorities of only those queues in the SPM that correspond to members of the ONS.

The initial logic level is established by AND-gates 41, which filter out (set to zero) the priority values of queues corresponding to output-nodes that are not members of the present ONS. The output of the final stage in the hierarchy is the prevailing queue's output-node index and priority value. Similarly, any implementation for determining the maximal value that comprises concurrent comparisons of more than two priority values in order

to accelerate the process may be applied (including for example a hierarchical tree of maximum-of-M units).

A similar maximal priority value circuitry to that described above may be implemented at the Grant Unit 27. However, it is to be understood that the method for derivation of the identity of the input-node or queue having the highest priority from among a collection of input-nodes or queues is not itself a feature of the present invention and other schemes may be equally well employed.

In the case that no requests were made by input-nodes for a specific output-node, the group of requests corresponding to that output-node will be empty. When such a condition is detected, no match is made for this output-node and it will remain available (unmatched) for succeeding iterations. Notwithstanding the above, if such a condition is detected for all members of the ONS, then no match is made for any of them, but they are all marked as unavailable so that none of them will participate as a member of the ONS in further iterations.

It is another task of the SMM 19 to maintain the sets of both available input-nodes and available output-nodes for the duration of each time-slot. These sets are updated for every match made by the Grant Unit 27. An input-node that has been granted (matched with) an output-node is disregarded in consecutive iterations. Similarly, an output–node matched to an input-node is registered by the SMM as unavailable for succeeding iterations within the same time slot, and will not be selected as a member of another ONS.

According to a preferred embodiment of the invention, the duration of the entire input-output-node matching process represents a "time slot". An "iteration" denotes a step within a specific time slot. Accordingly, a time slot is composed of one or more iterations. The time slot duration may be determined by one or more conditions, such as full matching of all

nodes, exhaustion of channels or a condition on the maximal number of iterations allowed per time slot.

Referring now to Fig. 5, there will be summarized the principal steps carried out by an algorithm executed by the scheduler 15. The matching of input-nodes with output-nodes is achieved by conducting a sequence of output-node contentions (iterations), in each of which unmatched input–nodes contend for a given subset of the unmatched output-nodes. At the end of each such iteration, input-output-node matches are established. These matches are accumulated to form a complete matching configuration at the end of the time slot. During any single time slot, data passes through the cross-connect in accordance with the matching decisions (configuration) established in the previous time slot. In such manner, no transmission dead-time is introduced.

In the first step of the algorithm, as noted above with reference to Fig. 2 of the drawings, each SPM produces a "request", based on the ONS and on the queue priorities maintained inside that SPM. This request consists of (a) the index of an output-node, which must be a member of the ONS, whose corresponding queue within the SPM has the highest priority value of all queues in the SPM corresponding to members of the ONS; (b) the priority value associated with the corresponding queue.

The collection of requests from the SPMs of all unmatched input-nodes serves as input to a second stage of the algorithm carried out by the Request Grouping Unit 26, which partitions the collection of requests into groups. As also explained above, each such group contains all requests made by input-nodes for one specific output–node. The requests are grouped by the Grant Unit 27 according to output-node identity and the output-nodes associated with each group are then concurrently matched with the input-node having the highest priority request in the respective group. The matches are accumulated and the matched input- and output-nodes are then removed from the sets of available input- and output-nodes.

The procedure is now repeated, as required, for the new sets of available nodes until expiry of the current time slot.

Fig. 6 shows a preferred algorithm for determining the request to be submitted by an SPM in the algorithm described above with reference to Fig 5. First, queues are grouped according to their corresponding output-node, and in each group, the queue having the highest priority is selected. Then zero priority is assigned to all selected queues whose corresponding output-nodes are not in the ONS, and the output-node whose selected queue has the highest priority is selected. A request is compiled containing the identity of the selected output-node and the priority of its corresponding selected queue.

It is possible for the invention to be applied to certain 'blocking' cross-connect fabrics in which the establishment of a channel may prevent (block) the establishment of further channels connecting nodes other than those connected by the established channel. If such a fabric is to be used, then upon the creation of a match and the allocation of the corresponding channel, the SMM will remove from both sets of available nodes those input- or output-nodes that were blocked by the allocated channel.

It is another task of the SMM to assure that the presented ONS's are in such composition and order to maximize efficiency and QoS provisioning.

An end-of-timeslot (EOTS) condition is determined by the SMM upon detecting the occurrence of any predetermined combination of events. The most preemptive such event is the 'satisfaction' case in which for all unmatched input-nodes, the priority of all queues corresponding to the set of unmatched output-nodes is zero. In such an event further iteration can yield no more matches and the time slot must be terminated. To allow for the detection of this event, each SPM provides the SMM with a signal or signals from which the SMM can infer a 'satisfaction' condition in that SPM.

Other examples of conditions that can be used by the SMM to determine an EOTS are: (a) exhaustion of cross-connect channels; (b) the duration of the time slot has exceeded a preset number of iterations or a preset amount of time; (c) the priorities of matches made during the time slot have accumulated to exceed a preset threshold, (d) a predetermined number of iterations; (e) an accumulated number of matches exceeds a predetermined threshold.

In the event of EOTS, the input-output-node matches accumulated during the time slot are passed to the cross-connect control circuitry, the sets of unmatched nodes are reset and a succeeding time slot is initiated.

The above technique may employ a pipelined implementation to accelerate the matching process (shortening time-slot duration). In this manner, different stages of the algorithm are carried out concurrently in separate stages of the architecture, with the output of a stage being fed to the input of its successor. Higher processing speed is gained at the expense of a constant latency derived from the pipeline stages.

The SMM can reduce the time slot duration by identifying output-nodes that are not requested by any input-node or by gathering statistical information based on which the Offered Nodes Sets are to be produced.

In the preferred embodiment, the algorithm performed by the scheduler performs the initialization steps of making a snapshot of queue priorities and defining sets of available input-nodes and available output–nodes each containing all input-nodes and output–nodes, respectively. However, it will be understood that the initialization can be performed independently such that the scheduler receives the snapshot and the sets of initially available input-nodes and available output–nodes. This can be used, for example, to shorten the time-slot duration by defining the sets of available nodes to initially contain only a (possibly random) subset of the nodes actually present in the switch.

Likewise, the processing of submitting one request for each input-node may be performed for all input-nodes concurrently. So too matching the output-node associated with each group with the input-node having the highest priority request in the respective group may be performed for all output-nodes in the ONS concurrently. Alternatively, these processes can be carried out in any desired serial manner.

The invention is also directed to an apparatus for real-time packet scheduling in high-rate, high port density, packet switched networks supporting QoS, comprising circuitry for locally determining packet scheduling at each input-node, according to information about priorities and available transmission resources at all other switch nodes which is collected over all said switch nodes in real-time.

The apparatus consists of a switch with a plurality of input-ports and output-ports, and switching control circuitry for controlling data transfer from input-ports to output-ports using assigned channels, and a cross–connect for the transmission of data through the channels. A scheduler controls the channel assignment process via designated control lines.

In the method claims that follow, alphabetic characters used to designate claim steps are provided for convenience only and do not imply any particular order of performing the steps.

CLAIMS:

1. A method for scheduling data packets transported from input-nodes to output-nodes via a cross-connect fabric having C channels, said data packets being associated with a set of N input-nodes each having a plurality of M queues each for queuing data packets for routing to a corresponding one of M output-nodes, said method comprising the steps of:

   (a) associating a priority value with each queue in each input-node, and making a snapshot of queue priorities,

   (b) receiving sets of available input-nodes and available output–nodes which may initially contain all input-nodes and output–nodes, respectively,

   (c) selecting a subset (ONS) of the set of available output-nodes,

   (d) submitting one request for each input-node, said request containing an identity of a requested output-node in the ONS and a corresponding priority value,

   (e) grouping requests according to the identity of the requested output-node,

   (f) matching the output-node associated with each group with the input-node having the highest priority request in the respective group,

   (g) accumulating matches and removing matched input- and output–nodes from the respective sets of available input- and output–nodes, and

   (h) repeating steps **(c)** to **(g)** as required.

2. The method according to Claim 1, wherein step **(d)** is performed for all input-nodes concurrently.

3. The method according to Claim 1 or 2, wherein step **(f)** is performed for all output-nodes in the ONS concurrently.

4. The method according to any one of the preceding claims, wherein step **(c)** is performed concurrently with steps **(d)** to **(f)**.

5. The method according to Claim 4, wherein step **(d)** is performed concurrently with steps **(e)** to **(f)**.

6. The method according to any one of Claims 1 to 5, wherein steps **(c)** to **(g)** are repeated until for each unmatched input-node, the priorities of all queues corresponding to the set of unmatched output-nodes are zero.

7. The method according to any one of Claims 1 to 5, wherein steps **(c)** to **(g)** are repeated for a predetermined number of iterations.

8. The method according to any one of Claims 1 to 5, wherein steps **(c)** to **(g)** are repeated for up to a predetermined time.

9. The method according to any one of Claims 1 to 5, wherein steps **(c)** to **(g)** are repeated until an accumulated value of the priorities of matched input-nodes exceeds a predetermined threshold.

10. The method according to any one of Claims 1 to 5, wherein steps **(c)** to **(g)** are repeated until an accumulated number of matches exceeds a predetermined threshold.

11. The method according to any one of Claims 1 to 5, wherein steps **(c)** to **(g)** are repeated until no more channels of the switching fabric are available to be allocated.

12. The method according to any one of Claims 1 to 5, wherein steps **(c)** to **(g)** are repeated until a logical combination is satisfied relating to:

- **i)** the priorities of all queues corresponding to the set of unmatched output-nodes are zero.
- **ii)** a predetermined number of iterations.
- **iii)** a predetermined time,
- **iv)** an accumulated value of the priorities of matched input–nodes exceeds a predetermined threshold.
- **v)** an accumulated number of matches exceeds a predetermined threshold . and

vi) no more channels of the switching fabric are available to be allocated.

13. The method according to any one of the preceding claims, wherein in step **(c)** the subset of available output-nodes is selected randomly to contain at most K output-nodes, where K is any integer between **1** and M.

14. The method according to any one of Claims 1 to 13, wherein in step **(c)** the subset of available output-nodes is selected in a sequential manner to contain at least two output-nodes.

15. The method according to any one of the preceding claims, wherein in step **(d)** the highest priority request in the respective input-node is determined by the following steps:

i) grouping queues according to their corresponding output-node,

ii) in each group, selecting the queue having the highest priority,

iii) assigning zero priority to all selected queues whose corresponding output-nodes are not in the ONS,

iv) selecting the output-node whose selected queue has the highest priority, and

v) compiling a request containing the identity of the selected output-node and the priority of its corresponding selected queue.

16. The method according to any one of the preceding Claims, wherein specified input- or output-nodes are blocked upon matching an output-node with an input-node and there is further included the step of removing the blocked input- or output-nodes from the respective sets of available input-nodes and available output-nodes.

17. A scheduler (15) for scheduling data packets transported from input-nodes (11) to output-nodes (12) via a cross-connect fabric (13) having C channels, said data packets being associated with a set of N input-nodes

each having a plurality of M queues each for queuing data packets for routing to a corresponding one of M output-nodes, said scheduler comprising:

an available input-nodes register (16) and an available output-nodes register (17) for storing data representative of respective sets of available input–nodes and available output–nodes,

an offer generator (18) coupled to the available output-nodes register for selecting a subset (ONS) of the set of available output-nodes,

a collection of source-port modules (SPMs) (25) each containing at least one register for storing a snapshot of queue priorities, said SPMs being coupled to the offer generator for submitting requests containing an identity of a requested output-node in the ONS and a corresponding priority value,

a request grouping unit (26) coupled to the SPMs for grouping requests according to the identity of the requested output-node,

a grant unit (27) coupled to the request grouping unit for matching the output-node associated with each group with the input-node having the highest priority request in the respective group, and

a match accumulator (28) coupled to the grant unit for accumulating matches and removing matched input- and output–nodes from the respective sets of available input- and output–nodes.

18. The scheduler according to Claim 17, being adapted to operate in accordance with the method of any one of Claims 2 to 16.

19. The scheduler according to Claim 17 or 18, being implemented in a packet scheduler for a communications network.

20. The scheduler according to Claim 17 or 18, being implemented in a multi-processor computer.

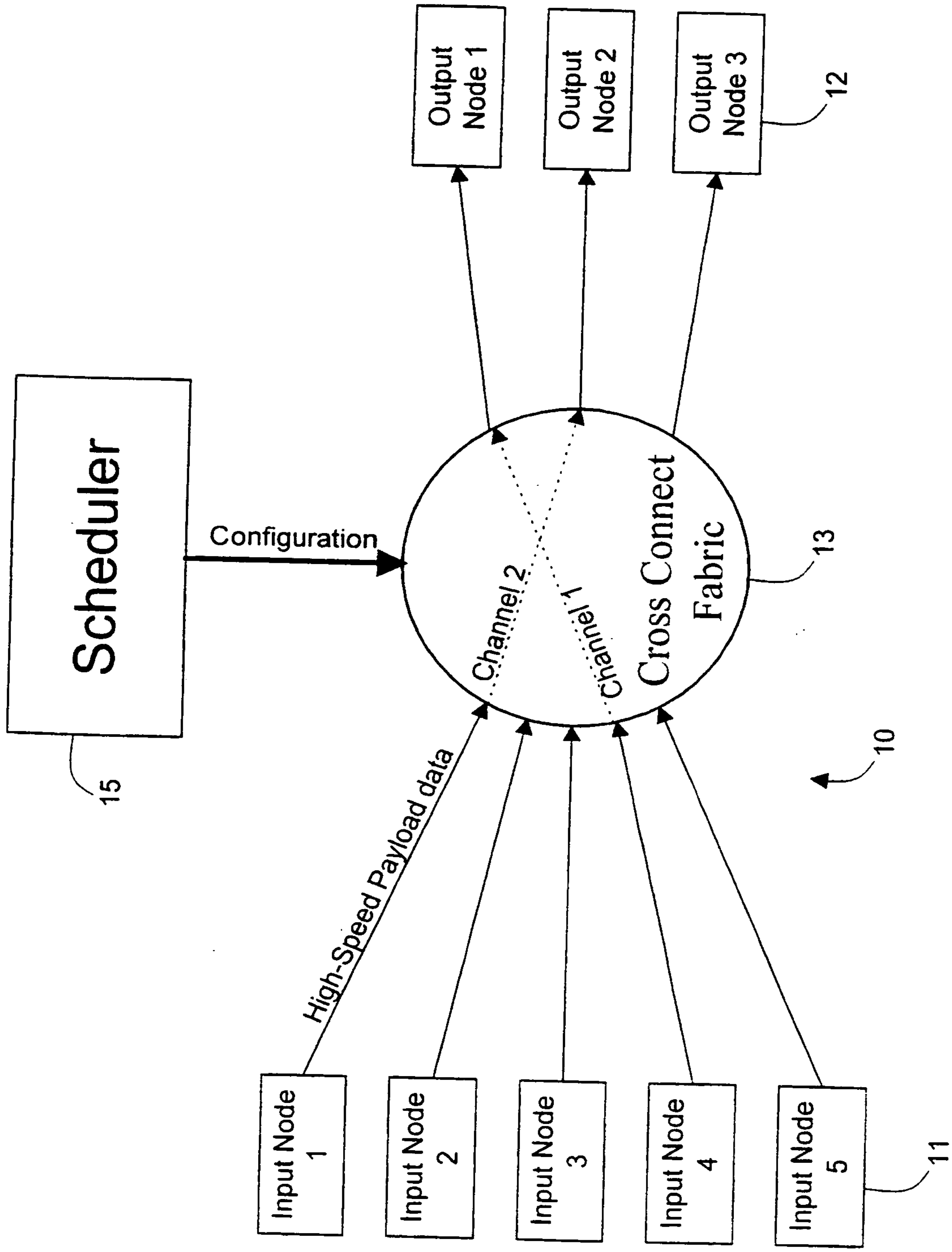

Scheduler
15
Configuration
Cross Connect Fabric
13
Channel 1
Channel 2
High-Speed Payload data
Output Node 1
Output Node 2
Output Node 3
12
Input Node 1
Input Node 2
Input Node 3
Input Node 4
Input Node 5
11
10

FIG. 1

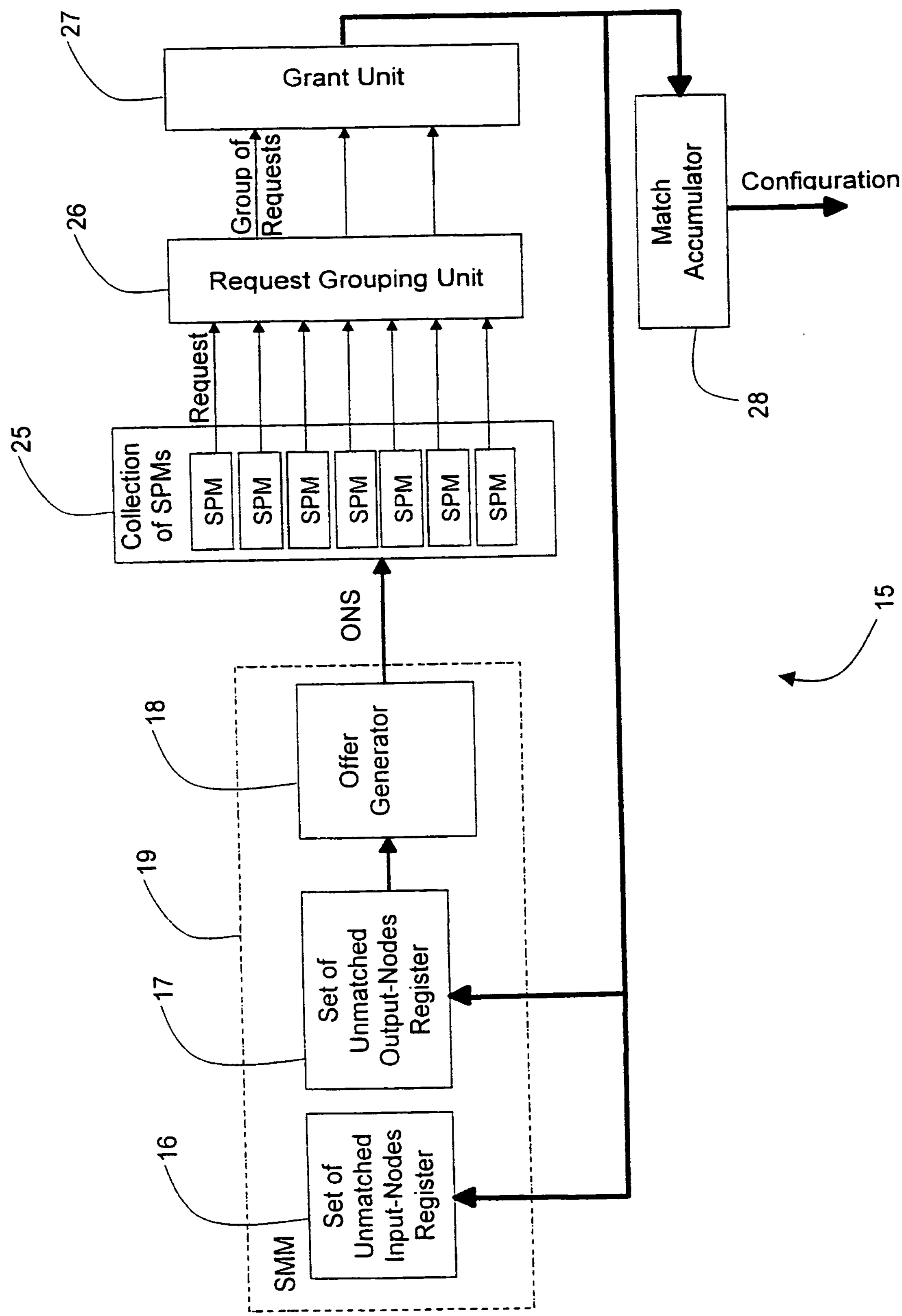

27
Grant Unit
Group of Requests
26
Request Grouping Unit
Request
25
Collection of SPMs
SPM
SPM
SPM
SPM
SPM
SPM
SPM
ONS
18
Offer Generator
19
17
Set of Unmatched Output-Nodes Register
16
Set of Unmatched Input-Nodes Register
SMM
Match Accumulator
28
Configuration
15

FIG. 2

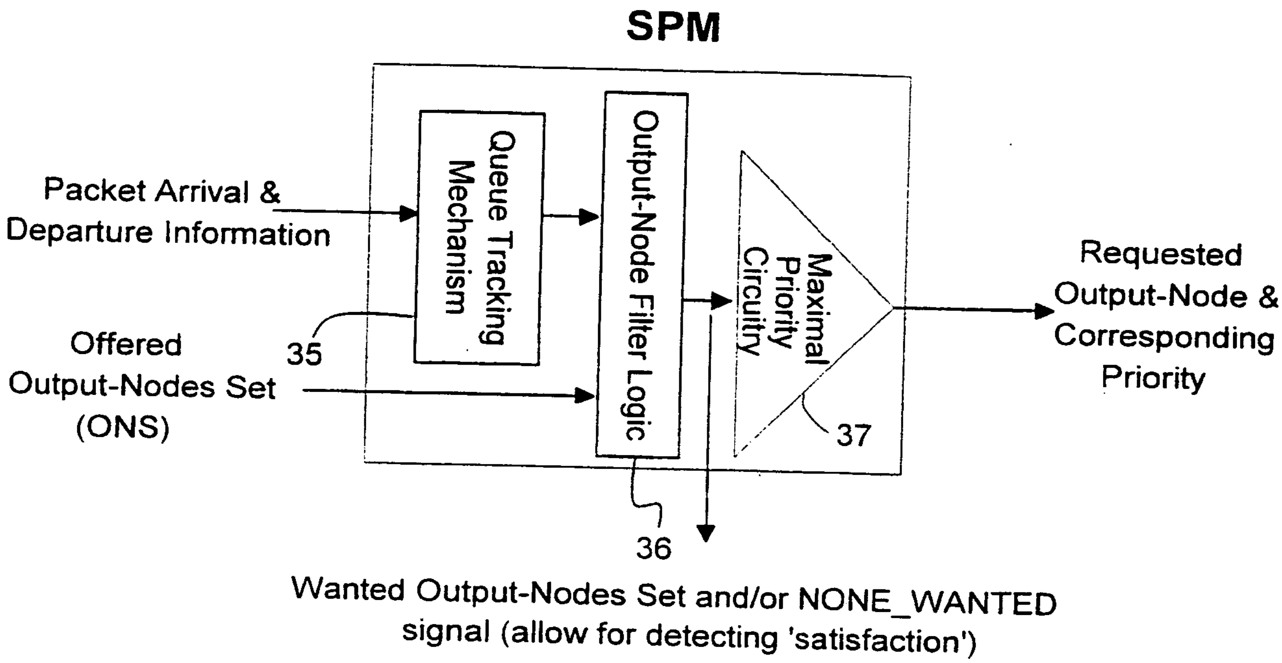

SPM
Packet Arrival & Departure Information
Queue Tracking Mechanism
Output-Node Filter Logic
Maximal Priority Circuitry
Offered Output-Nodes Set (ONS)
35
37
36
Requested Output-Node & Corresponding Priority
Wanted Output-Nodes Set and/or NONE_WANTED signal (allow for detecting 'satisfaction')

FIG. 3

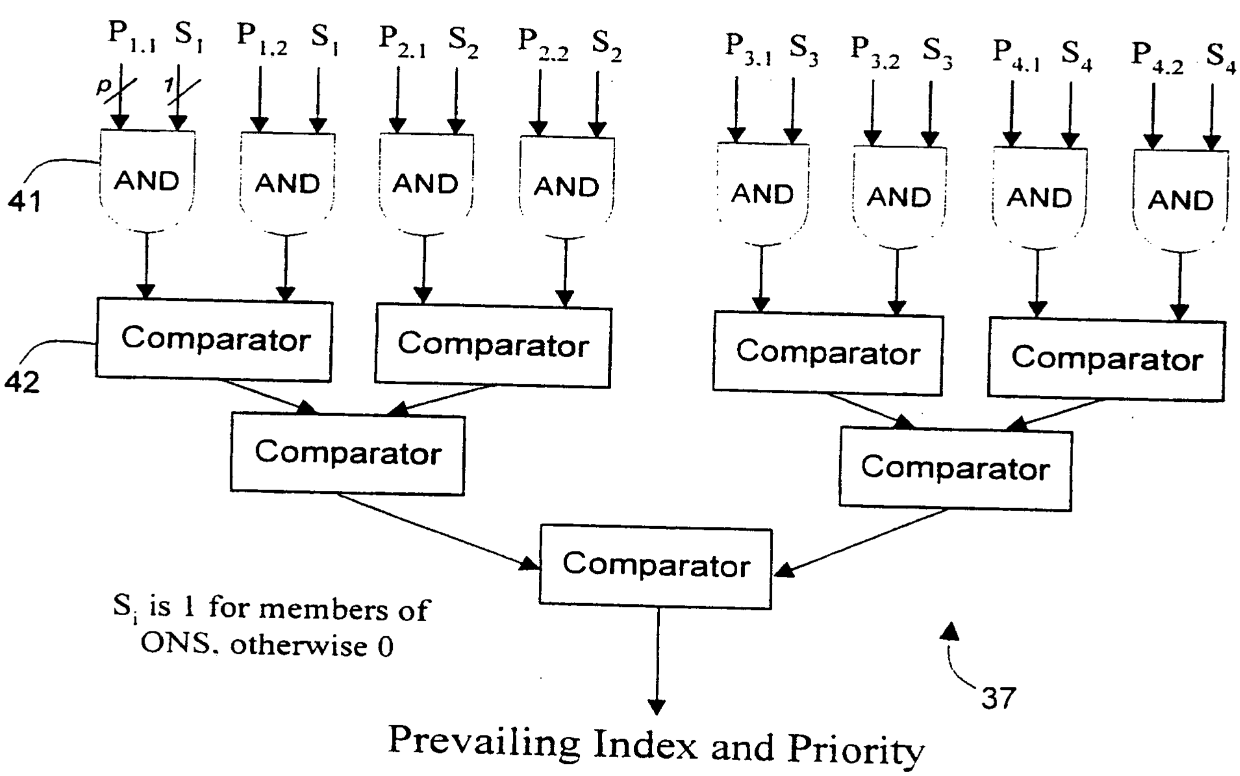

P1.1 S1 P1.2 S1 P2.1 S2 P2.2 S2 P3.1 S3 P3.2 S3 P4.1 S4 P4.2 S4
p
1
AND
41
Comparator
42
Si is 1 for members of ONS, otherwise 0
37
Prevailing Index and Priority

FIG. 4

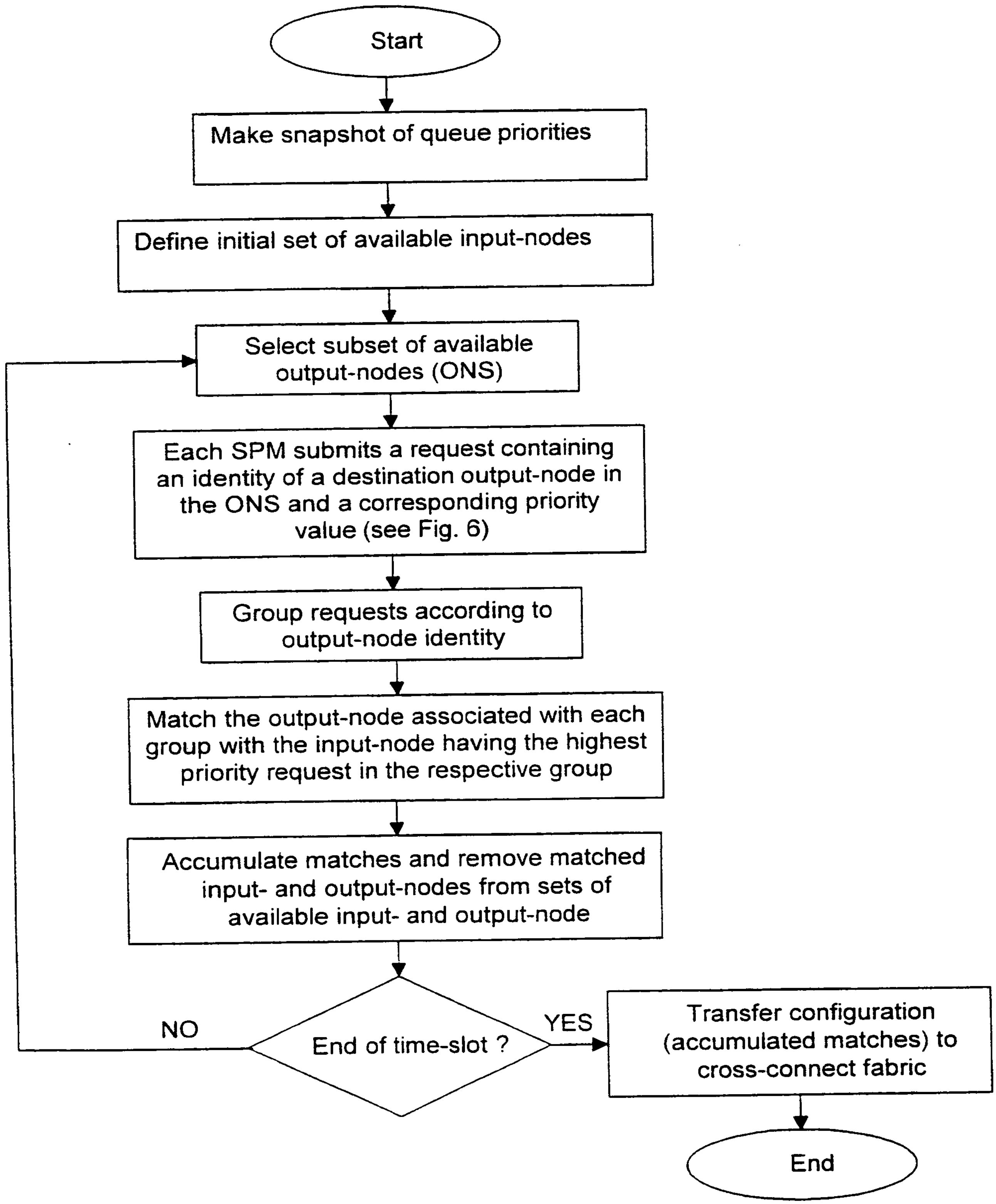

Start
Make snapshot of queue priorities
Define initial set of available input-nodes
Select subset of available output-nodes (ONS)
Each SPM submits a request containing an identity of a destination output-node in the ONS and a corresponding priority value (see Fig. 6)
Group requests according to output-node identity
Match the output-node associated with each group with the input-node having the highest priority request in the respective group
Accumulate matches and remove matched input- and output-nodes from sets of available input- and output-node
End of time-slot ?
NO
YES
Transfer configuration (accumulated matches) to cross-connect fabric
End

FIG. 5

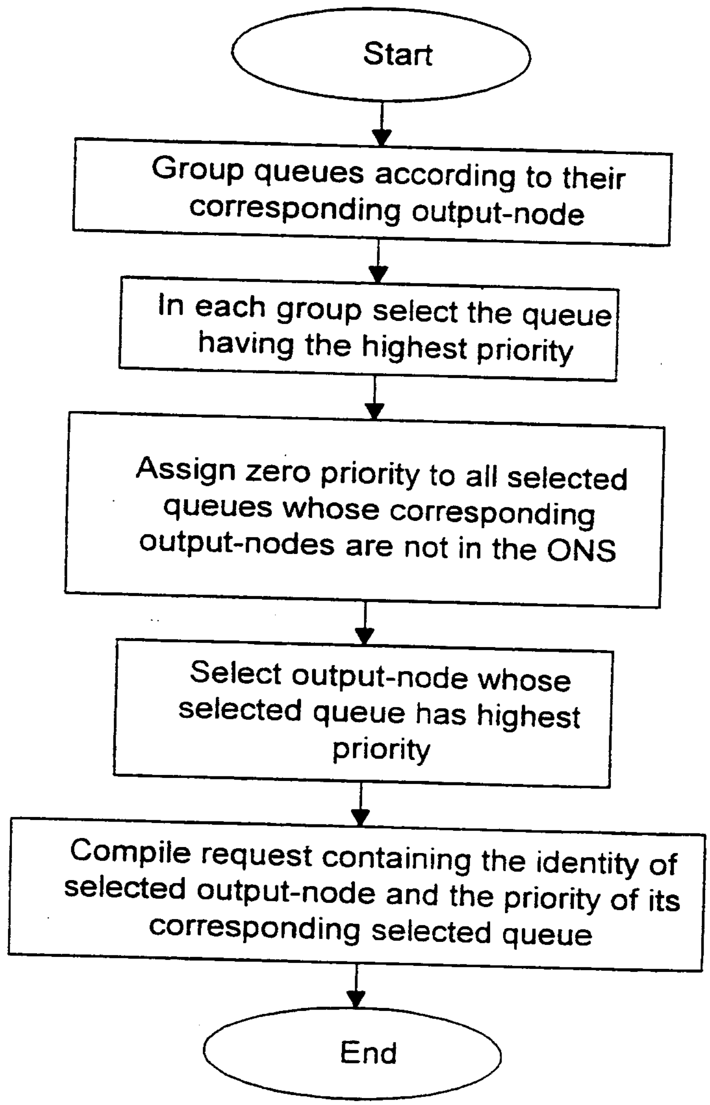

Start
Group queues according to their corresponding output-node
In each group select the queue having the highest priority
Assign zero priority to all selected queues whose corresponding output-nodes are not in the ONS
Select output-node whose selected queue has highest priority
Compile request containing the identity of selected output-node and the priority of its corresponding selected queue
End

FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IL 00/00581

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/56 H04Q11/04

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 923 656 A (DUAN HAORAN ET AL) 13 July 1999 (1999-07-13) column 3, line 48 -column 4, line 42	1-20
A	US 5 912 889 A (BELL ALAN G ET AL) 15 June 1999 (1999-06-15) column 2, line 40 -column 3, line 30; claim 1	1-20
A	WO 99 40754 A (CABLETRON SYSTEMS INC) 12 August 1999 (1999-08-12) abstract page 14, line 1 - line 11	11

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 November 2000	05/12/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Veen, G

4

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interna[illegible]l Application No
PCT/IL 00/00581

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5923656 A	13-07-1999	NONE	
US 5912889 A	15-06-1999	NONE	
WO 9940754 A	12-08-1999	AU 2598099 A	23-08-1999